

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Уральский государственный педагогический университет»
Институт математики, физики, информатики и технологий
Кафедра информационно-коммуникационных технологий в образовании

АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА СЕТЕВОЙ РАДИОСТАНЦИИ

*Выпускная квалификационная работа
бакалавра по направлению подготовки
09.03.02 – Информационные системы и технологии*

Исполнитель: студент группы ИСИТ-1501
Института математики, физики,
информатики и технологий
Феофилатов А.В.

Руководитель: д.п.н., заведующий кафедрой
ИКТО
Стариченко Б.Е.

Работа допущена к защите
«___» _____ 2019 г.
Зав. кафедрой _____

Екатеринбург 2019

Реферат

Феофилатов А.В. АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА СЕТЕВОЙ РАДИОСТАНЦИИ, выпускная квалификационная работа: 50 стр., рис. 30, библи. 40 назв., приложений 1.

Ключевые слова: сетевая радиостанция, информационная система, web приложение, фреймворк Django, база данных.

Объект разработки: автоматизированная информационная система сетевой радиостанции.

Цель разработки: спроектировать автоматизированную информационную систему сетевой радиостанции, а также с помощью программных средств реализовать рабочую версию в виде web приложения.

В рамках работы был произведен анализ существующих решений, среди Интернет радиостанций различного типа, а также проанализирован и отобран инструментарий для реализации и последующей эксплуатации автоматизированная информационная система сетевой радиостанции. В ходе анализа существующих интернет радиостанций были выделены основные преимущества предлагаемого решения. При анализе инструментария было выделено, что наиболее эффективным решением для реализации объекта разработки является backend фреймворк Django для реализации серверной стороны, и frontend фреймворк Vue.js, для реализации клиентской стороны.

Реализованное приложение удовлетворяет требованиям, указанным в техническом задании и прошло апробацию в Уральском Государственном Педагогическом Университете.

Оглавление

ВВЕДЕНИЕ.....	4
ГЛАВА 1. ОПРЕДЕЛЕНИЕ И МЕТОД РЕАЛИЗАЦИИ ИНТЕРНЕТ РАДИОСТАНЦИИ.....	6
1.1 ОПРЕДЕЛЕНИЕ ИНТЕРНЕТ-РАДИО	6
1.2 ВЫБОР ИНСТРУМЕНТАРИЯ И МЕТОДА РАЗРАБОТКИ.....	12
1.3 ТЕХНИЧЕСКОЕ ЗАДАНИЕ	20
ГЛАВА 2. РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ СЕТЕВОЙ РАДИОСТАНЦИИ.....	26
2.1 МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ СЕТЕВОЙ РАДИОСТАНЦИИ.....	26
2.2 РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ	32
2.3 РЕЗУЛЬТАТЫ АПРОБАЦИИ	45
ЗАКЛЮЧЕНИЕ	47
СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ.....	48
ПРИЛОЖЕНИЯ	52
<i>ПРИЛОЖЕНИЕ 1.....</i>	<i>52</i>

Введение

В настоящее время в современном мире, все чаще создаются музыкальные стриминговые сервисы для прослушивания музыки в режиме онлайн, чтобы иметь быстрый доступ к любой мировой музыке. В некоторых из таких музыкальных сервисов есть функция радио, когда нейросеть, автоматически подбирает музыкальные композиции по первой воспроизведенной песне. Но зачастую, случается так, что после первых музыкальных композиций, начинают преобладать треки другого жанра. Таким образом существует некоторая необходимость в онлайн радио ресурсе, который будет соответствовать одной тематике, и для удобства подбора треков, слушатели бы сами могли предлагать треки для прослушивания.

В статье Нестеровой Н.Г. описывается небольшой опрос в ходе которого были сделаны некоторые выводы «Опрос, на выявление востребованности Интернет-радио, был проведен и среди потенциальных Интернет-радиослушателей. Он дал следующие результаты: из опрошенных 54 студентов факультета журналистики и филологического факультета ТГУ не слушают Интернет-радио – 18, слушают регулярно – 30, слушают редко – 5, периодически – 1. Слушают студенты по Интернет-радио обычно эфирные записи. Отмечается интерес к конкретной радиостанции, к конкретному радиоматериалу...» [9]. На основании описанного исследования можно сделать вывод, что Интернет радио является востребованным среди молодежи.

Предмет разработки: автоматизированная информационная система сетевой радиостанции.

Цель разработки: разработать автоматизированную информационную систему Интернет радиостанцию

Задачи:

1. Произвести анализ технологий реализации сетевой радиостанции, выявить специфику автоматизированной информационной системы сетевой радиостанции.

2. Проанализировать и обосновать выбор аппаратно-программной платформы, определить структуру, объекты web-интерфейса автоматизированной информационной системы сетевой радиостанции.
3. В соответствии с техническим заданием смоделировать, спроектировать и реализовать автоматизированную информационную систему сетевой радиостанции.
4. Провести апробацию.

Глава 1. Определение и метод реализации Интернет радиостанции

1.1 Определение Интернет-радио

Для того чтобы понять, что такое сетевая радиостанция нужно определить, что такое радиостанция и веб приложение, а также рассмотреть примеры существующих аналогов.

Согласно определению взятому с информационного ресурса Википедия «разновидность беспроводной передачи информации, при которой в качестве носителя информации используются радиоволны, свободно распространяемые в пространстве» [13]. Радиоволны позволили человечеству передавать звук на дальние расстояния.

С помощью радиоволн, была создана такая технология как радиовещания. Радиовещание из взятого определения с ресурса Википедия «технология передачи по радио неограниченному числу слушателей речи, музыки и других звуковых эффектов или звуковой информации в радиозфире» [14].

Для того, чтобы передавать радио волны на дальние расстояния потребовалось строить радиостанции. На сайте Википедия дано такое определение радиостанции «один или несколько радиопередатчиков или радиоприёмников, или комбинация радиопередатчиков и радиоприёмников, включая вспомогательное оборудование, необходимые в определённом месте для организации службы радиосвязи» [16].

Для того чтобы можно было принимать радио сигнал, нужен радиоприемник. Как определено в источнике Википедия «устройство, соединяемое с антенной и служащее для осуществления радиоприёма, то есть для выделения сигналов из радиоизлучения» [15].

По многим факторам, в настоящее время типичная схема радио устарела по нескольким факторам [2]:

1. Содержать огромные радиовышки очень дорого, и часто не рентабельно.

2. Держать радиоприемник как отдельное устройство не выгодно по причине занятия лишнего места.
3. Нестабильность радиосигнала, ведущая к регулярным помехам.
4. Качество звука, проигрывает аудиозаписи переданной, в сети Интернет.

Чтобы упоминать эту категорию в дальнейшем материале нужно обозначить ее одним термином. Поэтому всю эту схему передачи радиосигнала, из радиостанций в радиоприемник в дальнейшем будем обозначать Традиционная схема радиопередачи.

В настоящее время радиоприемники встраивают в мобильные устройства, что на некоторое время продлило жизнь Традиционной схеме радиопередачи, но в 2018 году и эта тенденция сходит на нет [12] и в большинстве современных мобильных устройств, отсутствует радиоприемник для работы приема радиосигнала.

Но Радио не исчезло как понятие, а преобразовалось. Теперь Радио это аудиотрансляция в реальном времени. В связи с этим полностью поменялся подход к ведению радиотрансляций.

Большинство радиостанций ведет свою деятельность в сети Интернет. По данному из Википедии определению термин Интернет обозначают как «всемирную систему объединённых компьютерных сетей для хранения, обработки и передачи информации» [5].

Годом появления Интернета считается 1969, именно в этом году был запущен ARPANET [24] (от англ. Advanced Research Projects Agency Network), созданный по заказу американского агентства DARPA (англ. Defense Advanced Research Projects Agency) [17]. В дальнейшем по принципам использовавшимся в ARPANET, был создан NSFNET (от англ. National Science Foundation Network) [31] использовавшийся университетами и вычислительными центрами для связи друг с другом. И уже из NSFNET был создан привычный Интернет.

За 40 лет существования Интернета, он заместил собой все источники информации и в то же время собрал их все в себе. Теперь не нужно держать в

доме огромную библиотеку книг, огромное количество книг было оцифровано, и теперь их можно найти в сети Интернет. Огромная коллекция аудио и видео произведений также было оцифровано и теперь хранится в Интернете, где их можно найти в свободном доступе. Любые медиа, которые сегодня выходят, будь то фильмы, музыка, подкасты, новости и новые публикующиеся книги, одновременно с выходом на физических носителях также публикуются в Интернете. Радио как информационный источник не стало исключением и тоже перешло в Интернет.

Поначалу переносить схему работы Традиционной схемы радиопередачи, то есть трансляции аудио бесперебойно и в прямом эфире было не возможным по нескольким причинам [7]:

- дороговизна траффика
- недостаточная распространенность сети Интернет
- недоступность для большинства пользователей
- сложность при разработке веб-сайтов

При вышеперечисленных минусах радио в сети Интернет было нереализуемо, но взамен появилась альтернатива в виде технологии аудио подкастинга. Подкастинг по взятому определению из информационного ресурса Википедия «процесс создания и распространения звуковых или видеофайлов (подкастов) в стиле радио- и телепередач в Интернете (вещание в Интернете)» [11]. И такие сайты с подкастами можно определить, как первые Интернет радиостанции. И хотя подкастинг не транслировался прямым эфиром, это была актуальная альтернатива радиостанциям, которая доносила информацию в формате аудио, и через которую являлось возможным передавать актуальную информацию.

Как можно вывести из самого термина «Подкаст», компания Apple [23] является главным популяризатором Подкастов в мире. Эта компания благодаря своему сервису iTunes [27] перевела формат эфирного вещания в Интернет. Интернет подкасты стали доступны большому пласту людей, за счет удобного

метода продвижения подкастов в магазине iTunes. Новые подкасты выходили каждый день, их можно было загрузить на свой iPod [26] и слушать где угодно. Для примера на рис. 1 показан скриншот Домашней страницы iTunes Podcasts. Но даже в таком режиме был минус, который заключался в том, что без доступа к компьютеру с установленным iTunes и авторизованным профилем, являлось невозможным начать скачивать подкасты на свой iPod.

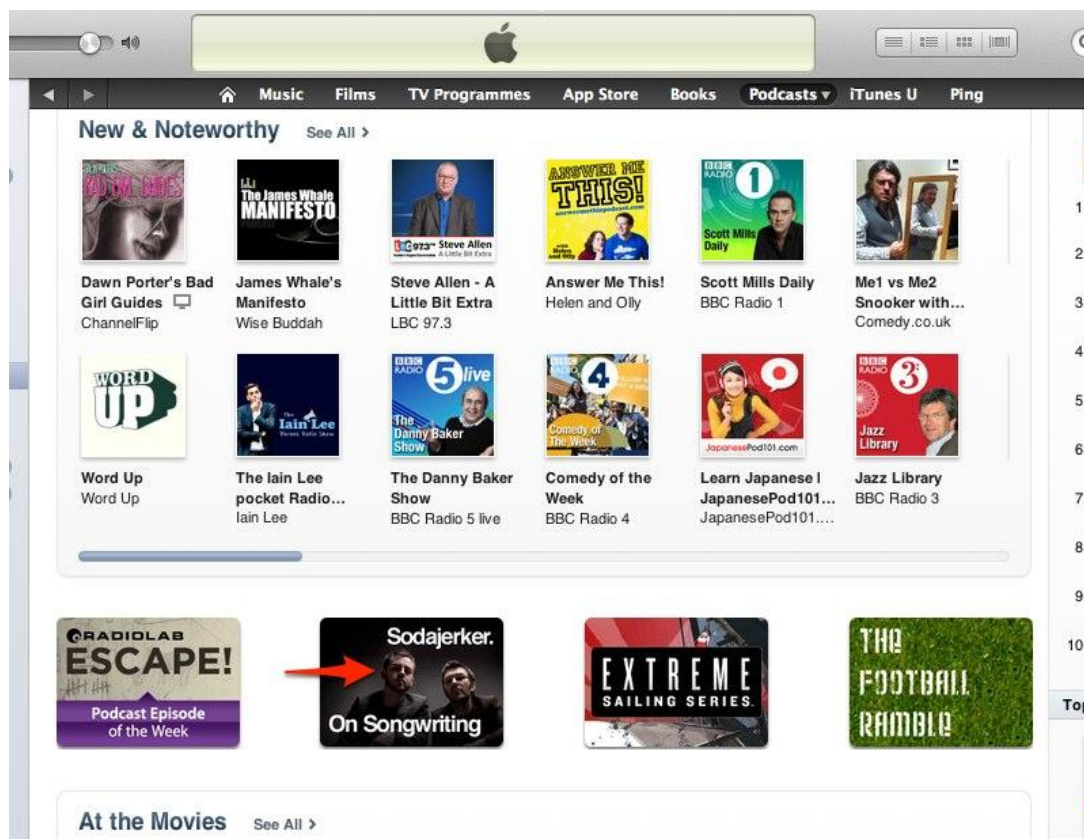


Рис. 1. Скриншот "Домашняя страница iTunes Podcasts" 2012 г.¹

Чем доступней становился Интернет, тем больше радиостанций перебрасывали свои ресурсы в веб пространство. Это оказалось гораздо выгодней [1]. Большие радиовышки сменились на относительно небольшие серверные шкафы. Но для работы многих Интернет-радиостанций требуется работник, или несколько работников, которые будут часто заниматься подбором плей-листа [10].

¹ https://sodajerker.com/wp-content/uploads/2012/04/itunes_feature.jpg

Таким образом есть некоторый запрос, который требует большей автоматизации Интернет-радиостанций, в аспекте удешевления эксплуатации.

Можно выделить несколько подходящих под определение автоматизированных сетевых радиостанций Интернет ресурсов.

Первым можно выделить известный стриминговый музыкальный ресурс Youtube Music [40]. Для примера на рис. 2 показан Главный экран Youtube Music. Он имеет большую картотеку медиафайлов и интегрирован напрямую с сервисом видео просмотра Youtube [39]. Youtube Music имеет режим радио, когда пользователь выбирает для прослушивания аудиозапись определенной тематики; при этом сервис Youtube Music, используя искусственную нейронную сеть [6], подбирает дальнейший плей-лист по первой воспроизведенной музыкальной аудиозаписи, а также исходя из предыдущих предпочтений.

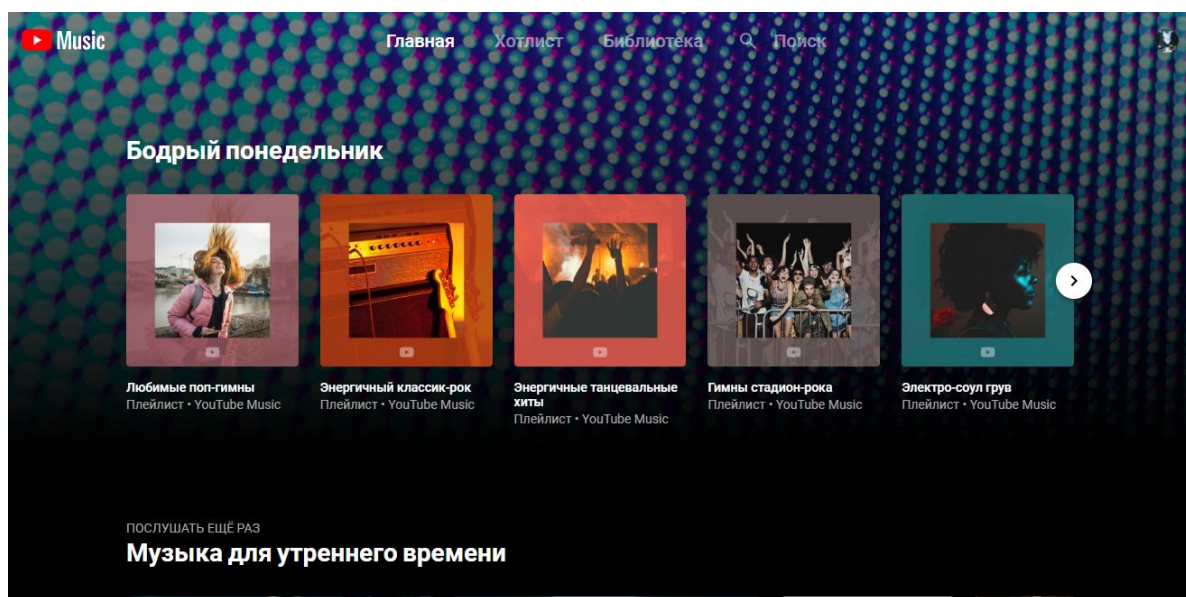


Рис. 2. Главный экран Youtube Music

Минус этого метода подбора треков в том, что каждая последующая проигрываемая аудиозапись может своим набором жанров повлиять на будущую генерацию плей-листа, чем порой сбивает тематику плей-листа в совершенно другое направление, что может не удовлетворять множество пользователей.

Также есть Российский аналог сервиса Youtube Music от компании Яндекс под названием Яндекс Музыка, который работает по тому же принципу, и имеет проблемы схожей специфики.

Другой тип подхода к автоматизации сетевой радиостанции можно найти на веб сайте Anison.fm. Например, на рис. 3 показана главная страница сайта Anison.fm. На этом веб сайте используется подбор музыкальных композиций с помощью выбора пользователей этого веб ресурса. Этот подход оправдан, так как пользователи сами могут выбирать что им лучше послушать, а достаточный онлайн сайта и обширная картотека оставляют некоторый элемент случайности воспроизведения музыкальных треков.

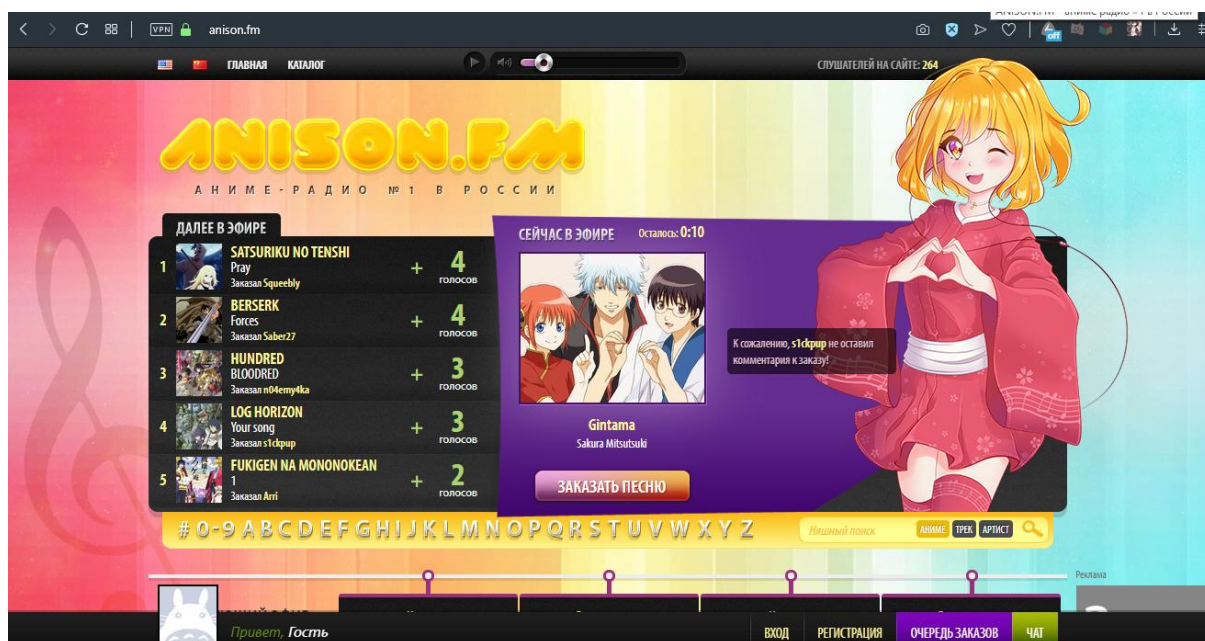


Рис 3. Главная страница сайта Anison.fm

Но в случае такой реализации возможно совершение нарушения закона если руководство или владельцы сайта не заключали договор с правообладателями контента на распространение материала опубликованного и используемого на данном веб ресурсе, потому что встраивание на сайт музыки без разрешений от правообладателя незаконно и несет за собой наказание от штрафа, до полной блокировки домена, на территории Российской Федерации [8].

На примере вышеприведенных Интернет радиостанций можно сделать вывод что автоматизированная информационная система сетевой радиостанции, это радиостанция не имеющая конкретного обслуживающего радиоведущего, а весь подбор композиций подвержен некоторому элементу случайности. Таким образом на основе приведенных ранее примеров можно выделить подход для дальнейшей разработки информационной системы. Выбор аудиозаписей для дальнейшего проигрывания будет лежать на пользователях, тем самым предоставляя некоторую свободу выбора и предпочтений. Реализация генерации списка аудиозаписей для проигрывания, будет как на веб сайте Anison.fm, т.е. все пользователи будут собирать общий плей-лист. Но реализация вывода аудиозаписей будет отличаться. Т.к. для распространения большинства контента требуется разрешение правообладателя, то вывод аудиозаписей будет осуществляться через Youtube видеоролики. Это позволит сэкономить физическую память на сервере, так как ссылки на видеоролики занимают значительно меньше места, чем исходные аудиозаписи, а также так как весь открытый контент на сайте Youtube публикуется с лицензией Creative Commons, то весь контент публикующийся с данной лицензией можно публиковать в свободном доступе, что решает проблему с нарушением закона по отношению к правообладателю.

1.2 Выбор инструментария и метода разработки

Разработка клиент серверного приложения состоит из двух основных направлений: Frontend (Фронтэнд) – клиентская часть и Backend (Бэкэнд) – серверная часть [25].

Backend веб-разработка состоит из бесчисленных задач. Например, защита API (интерфейсов прикладного программирования) [22] от сторонних атак, авторизация пользователей, упрощение безошибочного взаимодействия с базами данных [3] и маршрутизация URL-адресов (унифицированные указатели ресурсов) [34] в базы данных для получения информации, запрашиваемой

клиентами, и т.д. Базовые фреймворки [18], также называемые серверными веб-фреймворками, делают все эти задачи удобными и удобными для разработчиков.

Базовые платформы оцениваются по инструментам программирования, языкам и интерфейсам, которые они предлагают. Более того, разработчики предпочитают фреймворки, которые предлагают предварительно сконфигурированные инструменты и шаблоны, которые помогают им быстро отслеживать разнообразные задачи веб-разработки. Проще говоря, усовершенствованная базовая инфраструктура ускоряет скорость разработки, делая задачи менее трудоемкими для разработчиков. Продвинутый Backend не должен ограничиваться предоставлением структуры или просто методологии для разработки приложений, скорее он должен позволять разработчикам создавать совместимые платформы, которые могут выдерживать нагрузку в масштабе.

Существует бесчисленное множество фреймворков, которые разработчики используют для создания Backend веб-приложений:

- Ruby on Rails;
- Django;
- Node JS;
- Flask;
- PHP (Laravel, Symfony, Yii).

Остановимся подробнее на двух фреймворках, наиболее подходящих для реализации поставленной задачи. Далее пойдет обзор фреймворков Django и Node JS.

Django - это платформа с открытым исходным кодом для серверных веб-приложений на основе Python - одного из лучших языков веб-разработки [36]. Его основными целями являются простота, гибкость, надежность и масштабируемость [29].

Django имеет собственную систему именования для всех функций и компонентов (например, HTTP-ответы называются «представлениями»). Он также имеет панель администратора, с которой работать легче, чем в Lavarel или Yii, и другие технические функции, включая:

- простой синтаксис;
- собственный веб-сервер;
- архитектура ядра MVC (Model-View-Controller);
- поставляется со всем необходимым для решения общих вопросов;
- ORM (объектно-реляционное отображение);
- Библиотеки HTTP;
- Поддержка промежуточного программного обеспечения;
- Фреймворк для юнит-теста Python.

Кроме того, Django предоставляет динамический интерфейс CRUD (создание, чтение, обновление и удаление), настроенный с помощью моделей администратора и сгенерированный посредством самоанализа. CRUD используется для описания основных команд базы данных, что означает, что интерфейс облегчает просмотр, изменение и поиск информации [38].

Одна из главных целей Django - упростить работу для разработчиков. Для этого среда Django использует:

1. Принципы быстрой разработки, которые означают, что разработчики могут выполнять более одной итерации за раз, не запуская весь график с нуля.
2. Философия DRY - «Не повторяй себя» - означает, что разработчики могут повторно использовать существующий код и сосредоточиться на уникальном [20].

В результате, чтобы вывести проект на рынок, требуется намного меньше времени.

Безопасность также является высоким приоритетом для Django. Он имеет одну из лучших готовых систем безопасности и помогает разработчикам избежать распространенных проблем безопасности, в том числе:

- ClickJacking,
- межсайтовый скриптинг
- SQL-инъекции.

Django быстро выпускает новые патчи безопасности. Обычно это первый, который реагирует на уязвимости и предупреждает другие платформы.

С Django вы можете заниматься проектами любого размера и емкости, будь то простой веб-сайт или веб-приложение с высокой нагрузкой. Вот несколько причин использовать Django для своего проекта:

- Он полностью загружен дополнительными возможностями и масштабируется, поэтому вы можете создавать приложения, которые обрабатывают большой трафик и большие объемы информации;
- Кроссплатформенность, это означает, что ваш проект может разрабатываться на Mac, Linux или Windows;
- Он работает с большинством основных баз данных и позволяет использовать базу данных, которая больше подходит для конкретного проекта, или даже несколько баз данных одновременно.

Django проверен временем и действием. У него большое поддерживающее сообщество, доступ к которому осуществляется через многочисленные форумы, каналы и специализированные веб-сайты. Легко найти помощь, когда в коде есть проблематичная функция, и найти разработчиков, если ваша компания планирует основать следующий проект на Django.

Django имеет отличную документацию, лучшую из всех других фреймворков с открытым исходным кодом. И она по-прежнему

поддерживается на высоком уровне, обновляется вместе с новыми функциями и исправлениями, поэтому вы можете легко адаптироваться к изменениям.

Можно положиться на тот факт, что любые проблемы со структурой будут решены, как только они возникнут. Программное обеспечение постоянно обновляется и выпускаются новые пакеты, чтобы сделать работу с Django более удобной, чем она есть.

Django был впервые создан для создания веб-приложения для издателя газеты Lawrence Journal-World. Он отлично себя показал при работе с проектами с объемным текстовым контентом, медиафайлами и интенсивным трафиком.

Но издательская индустрия - не единственная сфера применения фреймворка. Django также используется для создания веб-сайтов электронной коммерции, медицинских и финансовых приложений для транспорта и бронирования, сайтов социальных сетей и многого другого. Вот некоторые из многих типов проектов, которые вы можете разработать с помощью фреймворка:

- Финансовые платформы с функциями анализа и расчета приблизительных результатов на основе персональных данных, толерантности к риску, вероятности достижения целей;
- Встроенные пользовательские системы CRM для внутренних данных;
- B2B CRM системы для управления коммуникациями между предприятиями;
- Платформы, которые облегчают общение между двумя сторонами, такими как бизнес и потребитель;
- Высоконагруженные системы бронирования или торговые платформы;
- Мобильные приложения для Android и iOS, поддерживающие веб-приложения;

- Системы оценки недвижимости;
- Системы управления документами;
- Платформы для решения юридических вопросов, таких как проверка условий ипотеки или статуса аренды.

Некоторые компании предпочитают основывать свои проекты на более чем одной структуре. Django также можно использовать для создания отдельных функций, таких как:

- Система электронной почты для отправки уведомлений пользователям;
- Система фильтрации с продвинутой логикой и динамически меняющимися правилами;
- Генераторы на основе алгоритмов;
- Инструменты анализа данных;
- Интерфейсы для управления инвестиционными фондами;
- Панели администратора;
- Системы верификации на основе фотографий;
- И другие функции, которые облегчают разработку CRM и B2B платформ, онлайн торговых площадок, систем бронирования и многого другого.

Альтернативой для фреймворка Django в качестве инструмента для разработки серверной части выступает фреймворк Node JS. Node JS - это среда выполнения JavaScript [28]. Под средой исполнения подразумевается инфраструктура для создания и запуска программных приложений. Для создания приложений на JavaScript, в этом случае. Для этого нужно рассмотреть, каковы версии определения Node JS.

Сама компания описывает Node как «среду выполнения Javascript, основанную на движке Chrome V8». Википедия утверждает, что Node.js – «это кроссплатформенная среда с открытым исходным кодом для выполнения кода»

[30]. Согласно Techtarget, это платформа разработки, предназначенная для создания серверных приложений. И PCMag говорит, что Node - это платформа с собственным веб-сервером для лучшего контроля. Этого, безусловно, достаточно, чтобы понять основную идею [37].

Краткое резюме будет следующим:

1. Node JS является серверной платформой и является бесплатной.
2. Работает на Windows, Linux, Mac OS и т. д.
3. Node использует JavaScript на сервере.

Как работает Node JS? Если взять простую задачу открытия файла на сервере, последовательность будет такой:

1. Задача идет в файловую систему
2. Система готова к следующим запросам
3. Когда файл открывается и читается, система отправляет содержимое клиенту

Другими словами, с Node вам не нужно ждать, и вы можете продолжить выполнение следующих задач. Это одна из причин, почему это так эффективно.

Еще возможности, доступные в Node js:

1. Создание динамического контента
2. Создание, открытие, чтение и удаление файлов на сервере.
3. Сбор и изменение данных в базе данных.

В итоге в ходе анализа был выбран фреймворк Django, из-за большого количества встроенных пакетов, позволяющих быстро разработать нужный для серверной работы функционал.

После выбора инструмента для реализации серверной части, требуется также проанализировать и выбрать инструменты для реализации клиентской части. Далее речь пойдет о трех программных средствах для frontend разработки: Angular, Vue и React.

Angular, разработанный Google, был впервые выпущен в 2010 году, что делает его старейшим из всех. Это основанный на TypeScript JavaScript-

фреймворк [21]. Существенный сдвиг произошел в 2016 году в связи с выпуском Angular 2 (и отказ от «JS» от первоначального названия - AngularJS). Angular 2+ известен как просто Angular. Хотя AngularJS (версия 1) все еще получает обновления, далее обсуждение будет сосредоточено на Angular. Последней стабильной версией является Angular 7, выпущенная в октябре 2018 года.

Vue, также известный как Vue.js, является самым молодым членом группы. Он был разработан бывшим сотрудником Google Эваном Ю в 2014 году. За последние несколько лет популярность Vue значительно изменилась, несмотря на то, что он не имеет поддержки крупной компании. Авторы Vue поддерживаются на Patreon. Vue 3, в настоящее время на стадии создания прототипа, планирует перейти на TypeScript [35].

React, разработанный Facebook, был первоначально выпущен в 2013 году. Facebook широко использует React в своих продуктах (Facebook, Instagram и WhatsApp) [33]. Текущая стабильная версия в 16.X, выпущенная в ноябре 2018 года.

Если нужно быстро изучить среду, то стоит выбирать между Vue.js и React.

На Vue.js легко перейти как пользователю Angular, так и React. Ведь здесь получается чистый html-код, знакомый всем разработчикам. Приемы и техники используются примерно те же, что и в Angular.

Если предполагается разработка крупного проекта, то стоит рассматривать Angular в качестве основы. Он обеспечивает максимальную гибкость и скорость рендеринга. Огромный опыт других разработчиков позволит решить вопросы, которые обязательно возникнут при работе над приложением. React окажется слишком объемным, а для Vue.js еще не существует большого количества гайд-лайнов.

Если к разработке в будущем будут привлекаться другие программисты, то Vue.js станет лучшим выбором. Ведь этот фреймворк не только прост для изучения, но и позволяет менять приложение без разрушения его архитектуры.

Если для проекта предусматривается многоступенчатое обновление и расширение функциональности в будущем, то стоит использовать Vue.js или React из-за превосходной обратной совместимости.

Таким образом, для разработки frontend стороны, был выбран Vue.js, по факторам легкого осваивания среды, из-за простоты ввода последующих обновлений и легкого перестраивания приложения без разрушения его архитектуры.

1.3 Техническое задание

1. Общие сведения.

1.1.Название организации заказчика.

Федеральное государственное бюджетное образовательное учреждение высшего образования «Уральский государственный педагогический университет». Институт математики, физики, информатики и технологий.

1.2.Название продукта

Автоматизированная информационная система сетевой радиостанции.

1.3.Назначение продукта

Прослушивание музыки в режиме онлайн с ПК

1.4.Категория пользователь.

Любые заинтересованные лица

1.5.Плановые сроки выполнения.

01.09.2019 – 20.05.2019

2. Характеристика области применения.

2.1.Процессы и структуры, в которых предполагается использование продукта.

Организации, занимающиеся организацией досуга либо личная инициатива пользователя.

2.2. Характеристика персонала (категории лиц, которые участвуют в разработке администратор, оператор, пользователь, разработчик, в скобках уровень подготовки).

- Разработчик (Знание/владение стеком технологий Python, Django, СУБД PostgreSQL, JQuery, Ajax, Vue.js);
- Администратор (Знание сетевых технологий, ОС MS Windows/Windows Server/Ubuntu-14.04, терминального доступа, средств удаленного администрирования);
- Пользователь (Обладание основами работы в сети Интернет).

3. Требование к продукту разработки

3.1. Требования к продукту в целом:

- Требуется спроектировать структуру базы данных, а также реализовать ее в рамках разрабатываемого веб приложения на выбранной системе управления базами данных.
- На сайте должно быть реализовано динамическое обновление для представления актуальных данных из базы данных.
- Требуется реализовать каталог аудиозаписей, с возможностью заказа аудиозаписей.
- Требуется реализовать систему голосования за выбранные ранее пользователями аудиозаписей, находящиеся в текущем списке аудиозаписей.
- Требуется реализовать базу профилей пользователей, а также систему управления профилями.

3.2. Аппаратные обеспечение.

3.2.1. Аппаратура для разработки

- 1GB ОЗУ;

- 64-разрядный процессор с тактовой частотой 1,4 ГГц;
- 1024x768 минимальное разрешение экрана;
- 300 МБ свободного места на жестком диске;

3.2.2. Аппаратное обеспечение эксплуатации.

- 2 GB ОЗУ
- 64-разрядный процессор с тактовой частотой 1,4 ГГц
- Совместимый с набором инструкций для архитектуры x64
- 5 ГБ свободного места на жестком диске

3.3. Программное обеспечение.

3.3.1. Разработка.

3.3.1.1. Системное ПО.

- Windows 7, 8, 10

3.3.1.2. Прикладное ПО.

- Python
- Django
- Django Rest Framework
- Браузер
- Postgre
- PyCharm
- pgAdmin
- Vue js
- Muse UI

3.3.2. Эксплуатация.

3.3.2.1. Системное ПО.

- Windows 7, 8, 10

3.3.2.2. Прикладное ПО.

- Браузер

3.4. Форматы входных/выходных данных, порядок их ввода в систему.

Вход:

Набор аудиозаписей

Запрос на странице сайта

Выход:

Список аудиозаписей

Выбранная аудиозапись

3.5.Порядок взаимодействия с другими системами.

Взаимодействие с медиатекой youtube

3.6.Меры защиты информации.

Приложение имеет открытый программный код

4. Требования к пользовательскому интерфейсу.

4.1.Общая характеристика пользовательского интерфейса.

- Минимальное разрешение экрана 1024*768
- Пользовательский интерфейс должен быть простым и минималистичным.
- Доступ к базе аудиозаписей должен быть реализован с помощью алфавитного поиска.
- Требуется реализация динамического изменения топа текущих выбранных пользователями аудиозаписей, в соответствии с получаемыми данными из базы данных.
- Требуется реализация автоматического переключения актуальной проигрываемой аудиозаписи.
- Основной функционал сайта, должен быть реализован в рамках одной страницы.

4.2.Размещение информации на экране, макет дизайна экрана (макеты, сколько экранов, столько макетов).

Header

LogOut

pic

Name song
Author song

Voices ☐

pic

Name song
Author song

Voices ☐

pic

Name song
Author song

Voices ☐

Video

Library

Рис. 4. Главный экран

A B C ...

pic

Name song
Author song

☐

pic

Name song
Author song

☐

pic

Name song
Author song

☐

pic

Name song
Author song

☐

pic

Name song
Author song

☐

Рис. 5. Выбор композиции через главный экран

24

LOGIN

PASSWORD

Войти

Рис. 6. Экран входа/Регистрации

5. Требования к документированию.

Разработка технического задания в соответствии с техническим заданием.

6. Порядок сдачи приёма продукта

В соответствии со сроками выполнения ВКР.

Глава 2. Разработка автоматизированной информационной системы сетевой радиостанции

2.1 Моделирование информационной системы сетевой радиостанции

Жизненный цикл автоматизированной информационной системы [4] сетевой радио станции будет происходить в несколько следующих этапов:

- проектирование;
- разработка;
- тестирование;
- эксплуатация;

В рамках проектирования системы разработчик собирает общие требования к информационной системе, так же в этом этапе происходит глубокий анализ программного обеспечения для разработки клиентской и серверной части информационной системы и подбор сетевого оборудования для функционирования как Интернет приложение. После сбора требований разработчику требуется построить функциональную модель будущей автоматизированной информационной системы сетевой радиостанции.

После этапа проектирования разработчик приступает к реализации информационной системы сетевой радиостанции. Разработчик распределяет все собранные требования при проектировании информационной системы, ранжирует по важности и трудозатратности, после чего формирует по этим требованиям, а также по требованиям оформленного ранее технического задания задачи по клиент серверной реализации заявленного продукта.

На этапе тестирования база данных информационной системы заполняется медиафайлами, а также отлавливаются все ошибки, не выявленные при разработке.

После чего проект вступает в эксплуатацию и становится доступным широкому кругу пользователей.

В ходе проектирования как показано на рис. 7 была построена важная для дальнейшей разработки функциональная модель информационной системы. Эта модель описывает основные части работы информационной системы, которые были выделены из требований, описанных в части технического задания.

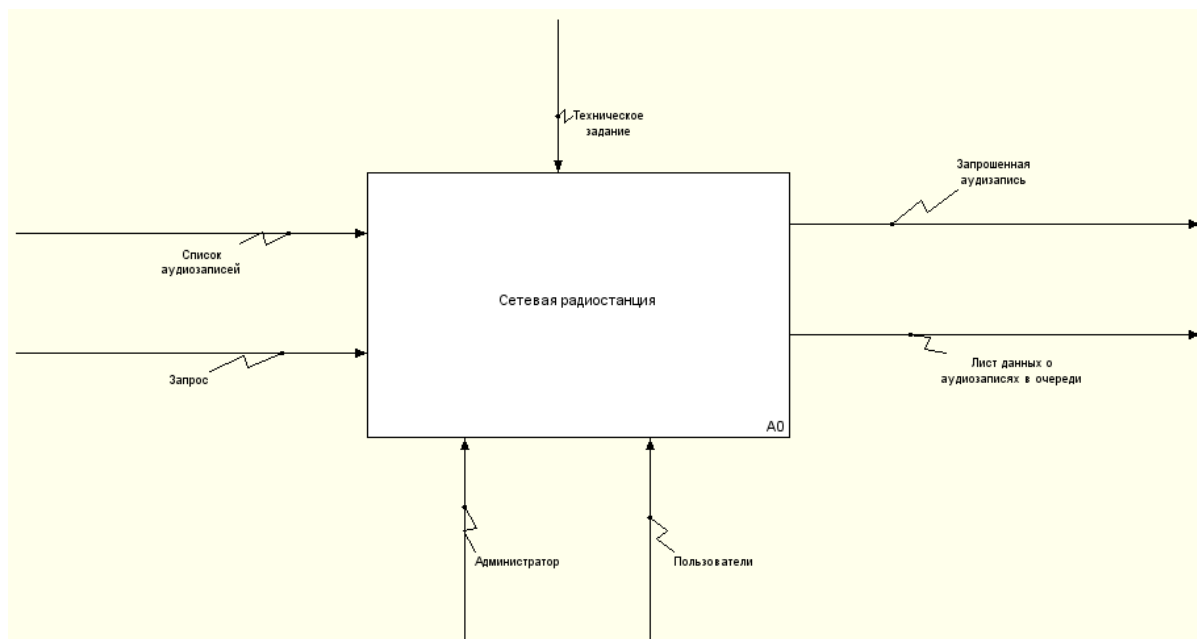


Рис. 7. Функциональная модель «Автоматизированной информационной системы сетевой радиостанции»

Работать с информационной системой будет два типа пользователей:

- Администратор проводящий минимальную степень модерирования информационной системы
- Регулярный пользователь

Разработчик занимается доработкой новых систем если такие понадобятся, разрабатывает новый функционал для информационной системы, отлавливает остаточные ошибки, если такие имеются. Администратор следит за состоянием системы, а также отвечает за наполнение базы данных. Пользователи же в свою очередь обеспечивают автоматизированность информационной системы сетевой радиостанции, за счет своих действий в пользовательском интерфейсе. Пользователи могут заказывать аудиозаписи для воспроизведения или могут голосовать за уже выдвинутые аудиозаписи, для их

более быстрого воспроизведения, таким образом, в этой системе администратор не занимается выставлением аудиозаписей на проигрывание.

На вход системы подаются список аудиозаписей, для дальнейшего их помещения в базу данных информационной системы и запрос, для дальнейшего получения по нему аудиозаписей.

Регламентирующим элементом информационной системы является техническое задание.

На втором уровне функциональной модели информационной системы, показанной на рис. 8, рассматриваются отношения серверной части и веб интерфейса.

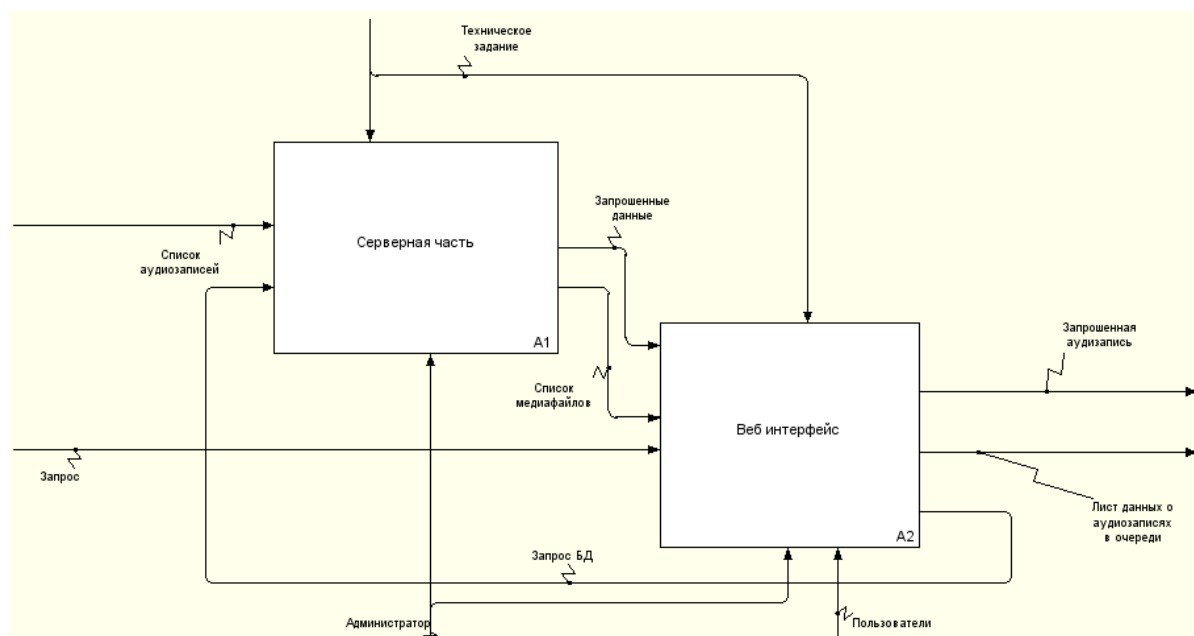


Рис. 8. Второй уровень функциональной модели «Автоматизированной информационной системы сетевой радиостанции»

На втором уровне было выделено два модуля:

- Серверная часть;
- Веб интерфейс

Администратор имеет доступ к обоим модулям. К серверной части для отслеживания ошибок по выдаче данных из базы данных, а также для добавления новых аудиозаписей для дальнейшего использования их в веб

интерфейсе. Администратор также имеет доступ к пользовательской базе чтобы в случае непредвиденных ситуаций и скорейшего из исправления.

Пользователь в свою очередь имеет доступ только к веб интерфейсу и может взаимодействовать только с итоговой очередью аудиозаписей и предложенной картотекой аудиозаписей.

В серверную часть поступают аудиозаписи, которые в последствии остаются в базе данных, для дальнейшей передачи их, на веб клиент по получению запроса на предоставление аудиозаписи.

Серверная часть, показанная на рис. 9, в детализации функциональной модели разделена на два модуля, это средства управления пользователями и средства управления медиафайлами.

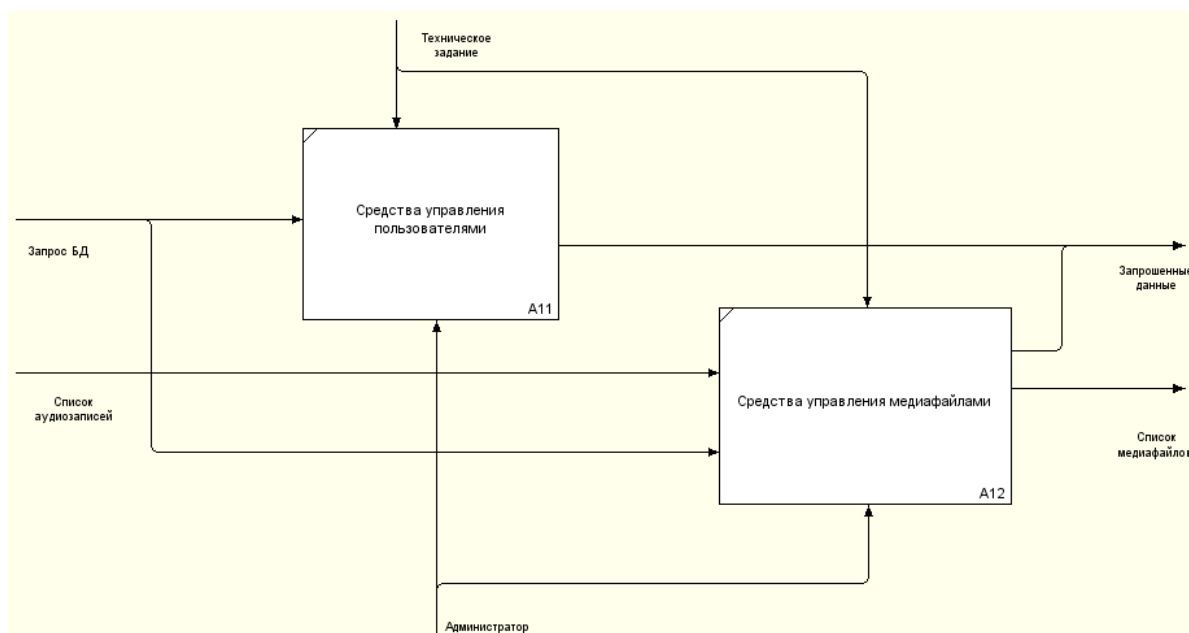


Рис. 9. Детализация серверной части функциональной модели «Автоматизированной информационной системы сетевой радиостанции»

На вход в средства управления пользователями идет запрос, который может как запрашивать данные о текущем пользователе для дальнейшей авторизации, так и запрос на создание нового профиля пользователя и дальнейшего его использования в веб клиенте. На выходе из средств управления пользователями идут итоговые сформированные данные, которые позволяют создавать ограничения в веб клиенте.

Средства управления аудиозаписями на вход принимают запросы и список аудиозаписей, который в дальнейшем будет обработан и добавлен в базу данных. На выходе отправляются данные, о текущем состоянии плейлиста, а также аудиозаписи для воспроизведения на веб клиенте. Доступ к средствам управления пользователями и средствам управления аудиозаписями имеет только администратор, в целях отлавливания непредвиденных ошибок и для безопасности базы данных.

В детализации веб интерфейса функциональной модели «Автоматизированной информационной системы сетевой радиостанции» показанной на рис. 10 рассматриваются отношения модулей клиентского обработчика данных и пользовательского интерфейса.

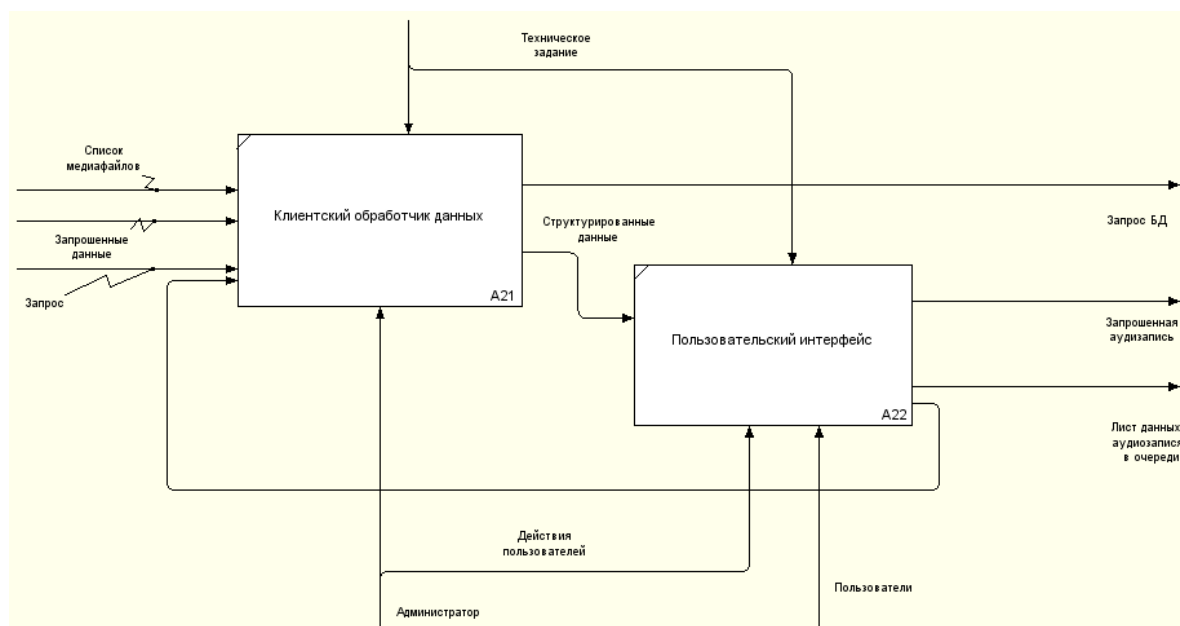


Рис. 10. Детализация веб интерфейса функциональной модели «Автоматизированной информационной системы сетевой радиостанции»

Администратор имеет доступ к обоим модулям веб интерфейса, для отладки ошибок и расширения существующего функционала веб клиента. Пользователь может взаимодействовать только с пользовательским интерфейсом откуда и отправляются все запросы.

Клиентский обработчик данных, это основной обработчик данных на веб клиенте. Клиентский обработчик данных получает все данные с серверной

части и обрабатывает их для дальнейшего представления в пользовательском интерфейсе. Клиентский обработчик данных получает на вход четыре источника данных:

- список данных о заказанных аудиозаписях;
- возвращенные по запросу данные;
- запрос данных с сервера;
- действия пользователей.

Список данных о заказанных аудиозаписях получается, для дальнейшего представления в пользовательском интерфейсе. Возвращенные по запросу данные, это данные которые клиентский обработчик данных запрашивал у серверной части и обрабатываются для автоматизации информационной системы. Запрос данных с сервера отправляется из серверной части для обработки данных на сервере и мгновенной перезаписи этих данных в базе данных. Действия пользователей обрабатываются для дальнейшей перерисовки пользовательского интерфейса и подстраивания под актуальные данные из базы данных. Также на основе действий пользователей формируются дальнейшие запросы, отправляемые в дальнейшем на серверную часть и активирующие передачу актуальных данных из серверной части. На выходе клиентского обработчика данных отправляются запросы в серверную часть, которые в зависимости от направленности активируют ту или иную отправку данных с серверной части.

Пользовательский интерфейс является основным рабочим местом для регулярных пользователей. Пользовательский интерфейс принимает на входе данные из клиентского обработчика данных и выводит их для пользователя в конечном виде. На выходе отправляются данные о пользовательских действиях, для дальнейшей генерации ответной реакции на действия пользователей. Также на выходе получается итоговая аудиозапись, которую прослушивают все пользователи.

2.2 Реализация информационной системы

Реализация информационной системы состояла из следующих пунктов:

1. Установка необходимого программного обеспечения.
2. Разработка клиент-серверной реализации информационной системы.
3. Развертывание реализованного приложения на сервере.
4. Описание результатов разработки.

Для реализации сетевой радиостанции было использовано заявленное в техническом задании программное обеспечение. Серверная часть была разработана на фреймворке Django. Django, базируется на языке Python и позволяет реализовать API для удобного вывода данных из базы данных.

Язык Python скачивается с официального сайта разработчиков Python. На главной странице сайта Python, нужно нажать кнопку Download. После перехода на страницу выбора версий для скачивания. Нужно найти актуальную версию Python и нажать кнопку Download Python 3.x.x. После скачивания, нужно открыть полученный файл и установить в нужную директорию.

Для удобства работы с языком Python была установлена интегрированная среда разработки PyCharm [32] от компании JetBrains. PyCharm значительно упрощает разработку приложений на языке Python. PyCharm распространяется в двух версиях: Community Edition и Professional Edition. Последняя является платной и поддерживает не только Python, но и большинство популярных библиотек и фреймворков для него, а также поддерживает контекстные подсказки для многих других языков, которые можно использовать вместе с python для веб-разработки. Чтобы скачать PyCharm, нужно зайти на официальный сайт JetBrains, перейти в раздел PyCharm и нажать кнопку Download, после чего выбрать подходящую версию, скачать и установить.

В PyCharm Professional можно автоматически загрузить фреймворк Django и как показано на рис. 11, создать проект в виртуальном окружении [19]. Но если нет PyCharm Professional, нужно зайти на официальный сайт Django

найти инструкцию по установке Django, и создать вручную Django проект в виртуальном окружении.

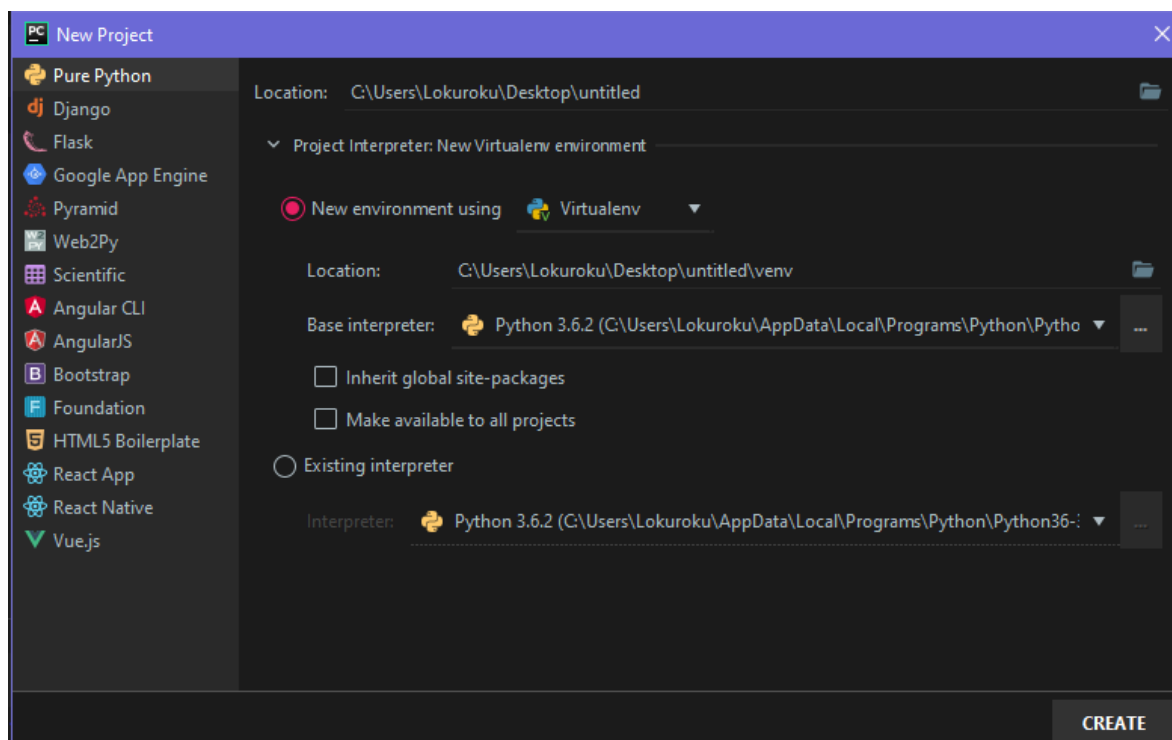


Рис. 11. Экран создания нового проекта в среде PyCharm Professional

После запуска проекта стандартный адрес локального сервера <http://127.0.0.1:8000/>, по этому адресу будет производиться отправка запросов, для получения данных с сервера.

Для того чтобы реализовать API нужно установить библиотеку Django rest framework. Метод установки, а также документация описаны на официальном сайте разработчика. После установки, потребуется подключить библиотеку в проект, делается это в настройках проекта, в файле “settings.py”.

Для обработки клиентской части будет использоваться frontend фреймворк. В нем будет описываться пользовательский интерфейс с помощью HTML, CSS и JavaScript скриптов. Процесс установки и создания проекта Vue описан на официальном сайте разработчика.

В качестве системы управления базой данных в проекте будет использоваться PostgreSQL. Для того чтобы установить эту систему управления

базой данных надо зайти на официальный сайт PostgreSQL, скачать дистрибутив установки и проинсталлировать PostgreSQL в систему. Также к PostgreSQL прилагается панель администрирования pgAdmin.

Разработка в Django происходит модульно, т.е. описываются отдельные элементы в различных модулях, после чего Django собирает весь проект воедино.

Все компоненты Django подключаются через python файл “settings.py”. Подключение базы данных происходит тоже через этот файл, как продемонстрировано на рис. 12.

```
4
5
6 DATABASES = {
7     'default': {
8         'ENGINE': 'django.db.backends.postgresql_psycopg2',
9         'NAME': 'radioapp',
10        'USER': 'postgres',
11        'PASSWORD': 'root',
12        'HOST': 'localhost',
13        'PORT': '5432',
14    }
15 }
```

Рис. 12. Подключение базы данных в Django

Для того чтобы подключить Django Rest Framework, как показано на рис. 13, в проект Django также требуется объявить его в файле “settings.py”. После этого станет доступным объявление сериализаторов.

```

4
5
6 REST_FRAMEWORK = {
7     'DEFAULT_PERMISSION_CLASSES': (
8         'rest_framework.permissions.IsAdminUser',
9         'rest_framework.permissions.AllowAny',
10        'rest_framework.permissions.IsAuthenticated',
11    ),
12    'PAGE_SIZE': 10,
13    'DEFAULT_AUTHENTICATION_CLASSES': (
14        # 'rest_framework_jwt.authentication.JSONWebTokenAuthentication',
15        'rest_framework.authentication.TokenAuthentication',
16        'rest_framework.authentication.BasicAuthentication',
17        'rest_framework.authentication.SessionAuthentication',
18    ),
19    'EXCEPTION_HANDLER': 'rest_framework_json_api.exceptions.exception_handler',
20    'DEFAULT_PAGINATION_CLASS':
21        'rest_framework_json_api.pagination.PageNumberPagination',
22    'DEFAULT_PARSER_CLASSES': (
23        'rest_framework_json_api.parsers.JSONParser',
24        'rest_framework.parsers.FormParser',
25        'rest_framework.parsers.MultiPartParser'
26    ),
27    'DEFAULT_RENDERER_CLASSES': (
28        'rest_framework_json_api.renderers.JSONRenderer',
29        'rest_framework.renderers.BrowsableAPIRenderer',
30    ),
31    'DEFAULT_METADATA_CLASS': 'rest_framework_json_api.metadata.JSONAPIMetadata',
32 }

```

Рис. 13. Объявление Django Rest Framework в настройках Django

Как продемонстрировано на рис. 14, процесс описания таблиц баз данных в Django происходит через модели. В коде через класс описывается модель, ее поля и типы этих полей после чего встроенная в Django ORM (Object-Relational Mapping, рус. объектно-реляционное отображение) преобразует эту модель в таблицу базы данных.

```

2
3
4 class Song(models.Model):
5     video = EmbedVideoField()
6     name = models.CharField(verbose_name="Название", max_length=300)
7     author = models.CharField(verbose_name="Автор", null=True, max_length=300)
8     duration_video = models.IntegerField(verbose_name="Длительность", null=True, default=0)
9
10    def __str__(self):
11        return "{} - {}".format(self.name, self.author)
12
13    class Meta:
14        verbose_name = "Песня"
15        verbose_name_plural = "Песни"

```

Рис. 14. Объявление модели Song для хранения медиафайлов

Чтобы перевести модель в таблицу базы данных, существуют миграции. Сначала создаются миграции командой `python manage.py makemigrations`, после чего модели мигрируют в базу данных с помощью команды `python manage.py migrate`.

Для вывода итоговых данных из базы данных в JSON формат объявляются сериализаторы. Сериализаторы создаются на основании ранее созданных моделей. Для начала нужно импортировать модель, после чего вызвать ее в методе Meta текущего класса, после чего следует прописать сериализуемые поля.

```
5
6
7     from rest_framework import serializers
8     from backend.models import Song
9
10
11 class SongSerializers(serializers.ModelSerializer):
12
13     class Meta:
14         model = Song
15         fields = ("video", "name", "author", "duration_video", "id", )
16 }
```

Рис. 15. Объявление сериализатора на основе модели Song

Формирование и отображение итоговых данных строится в модуле View.

В модуле View объявляются классы, методы которых будут содержать обработчики данных полученных из базы данных, по запросу, присланному из клиентского сервера. Для собрания итоговых данных строится класс, в котором получают данные из базы данных и в зависимости от прописанного алгоритма производится действие с базой данных.

```
2
3
4
5 class SongView(APIView):
6     permission_classes = [permissions.IsAuthenticated, ]
7     def get(self, request):
8         letter = request.GET.get("letter")
9         if (letter == "0-9"):
10             songs = Song.objects.filter(name__regex=r'\d*').order_by('name')
11         else:
12             songs = Song.objects.filter(name__istartswith=letter).order_by('name')
13         serializers = SongSerializers(songs, many=True)
14         return Response({'data': serializers.data})
```

Рис. 16. Объявление view класса для модели Song с методом get

Для того чтобы получить итоговые данные в Django используется вызов через URL. URL открывает прописанный в строке адрес, пример показан на

рис. 17, используя встроенный шаблонизатор. Django Rest Framework тоже использует URL, для получения и возврата данных. При вызове URL привязанного к определенному View классу возвращаются или получаются данные в формате JSON.

```
urlpatterns = [  
    path("songs/", SongView.as_view()),
```

Рис. 17. Объявление view класса в url путях

В свою очередь вызовы этих путей производятся с локального сервера для фронтэнда Vue.js.

Фронтэнд фреймворк отвечает за всю отрисовку пользовательского интерфейса, что позволяет разделить разработку frontend блока от backend. Такой подход позволяет независимо работать с этими двумя частями что значительно ускоряет процесс разработки.

Большая часть разработки во фреймворке Vue.js происходит в файлах компонентах типа Vue, эти файлы компоненты состоят из трех частей: template, scripts и style. Блок template является шаблонизатором, в этом блоке пишется Html разметка, также там применяются JavaScript скрипты и стили CSS. Блок scripts содержит все скрипты текущей Vue компоненты. С помощью JavaScript реализуется реактивность данных в данном фреймворке.

При входе на ресурс, как показано на рис. 18, пользователя встречает стартовый экран, для всех пользователей интерфейс идентичен.

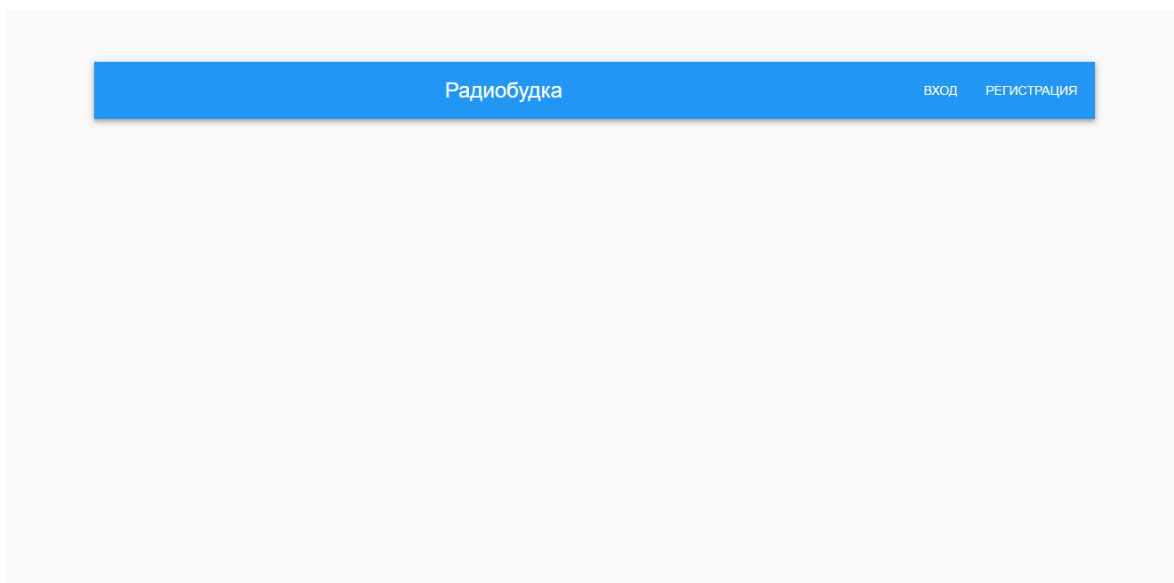


Рис. 18. Стартовый экран реализованного приложения.

Для дальнейшего пользования веб-сайтом следует нажать кнопку «Вход» и ввести предварительно заведенные данные. Система профилей реализована для ограничения количества голосов от пользователей, т.к. если не реализовать такую систему пользователи в дальнейшем смогут неограниченно выдвигать аудиозаписи из библиотеки, а также неограниченно голосовать за аудиозаписи, выдвинутые на голосование.

При открытии экрана авторизации, как показано на рис 19, система предлагает ввести логин и пароль, которые предварительно должны были быть созданы пользователем. После ввода учетных данных для входа в систему требуется нажать кнопку «Войти», после чего система в зависимости от корректности введенных данных выдаст одно из двух уведомлений.

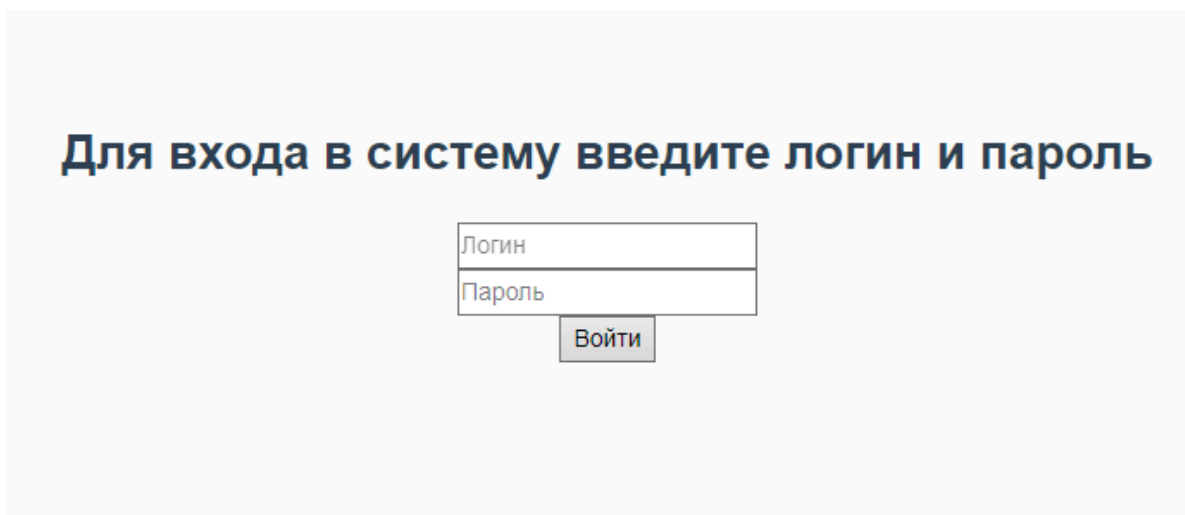


Рис. 19. Экран авторизации в системе

При попытке ввода не корректных данных перед пользователем всплывает сообщение об ошибке, как показано на рис. 20, в котором говорится что введенные данные не корректны. В таком случае пользователь должен заного ввести данные, пока они не будут корректными.

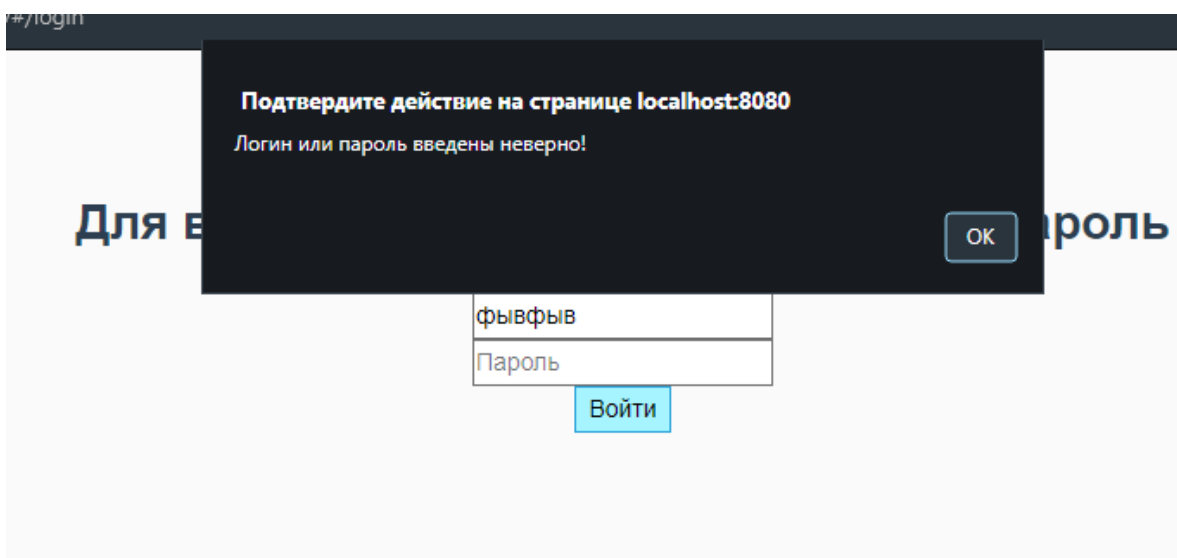


Рис. 20. Уведомление о неудачной попытке входа в систему

При вводе корректных данных, как показано на рис. 21, пользователю показывается сообщение об успешной попытке входа. Далее остается подтвердить действие нажатием кнопки **Ок** на всплывающем уведомлении, после чего система вернет пользователя на главный экран, с доступом к основному наполнению веб сайта.

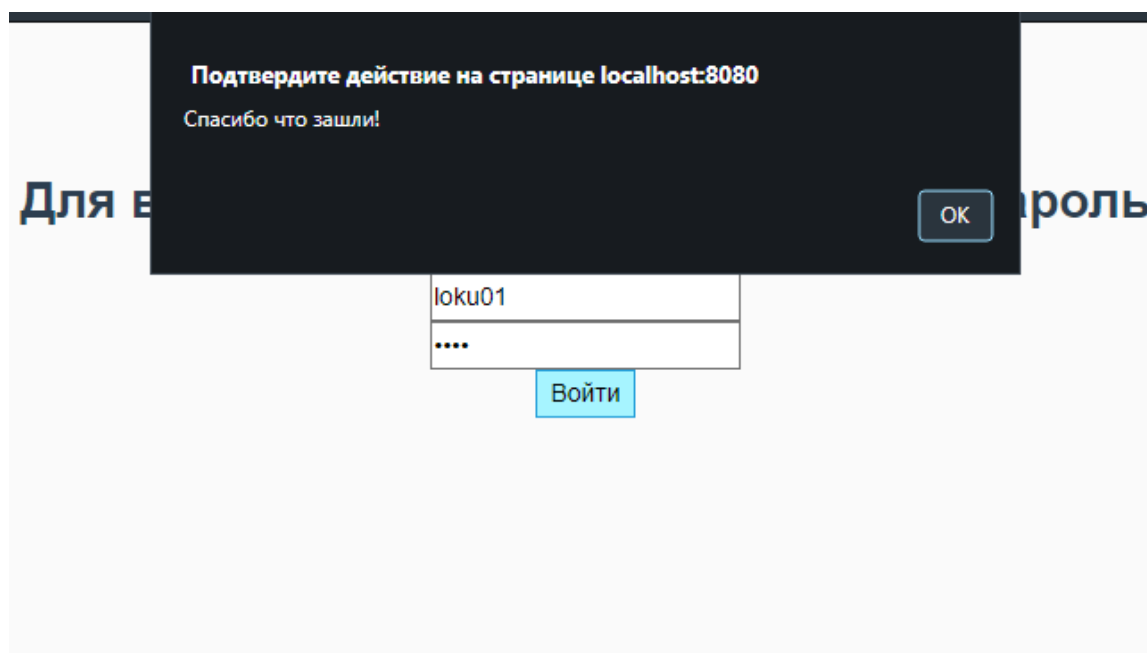


Рис. 21. Сообщение об успешном входе в систему

Также как показано на рис.22, в данном веб приложении реализована система регистрации, при которой создается новый пользователь, данные которого можно использовать для авторизации в системе.

The registration form is displayed on a light gray background. It features three lines of instructional text: "Для того чтобы зарегистрироваться", "введите логин и пароль,", and "затем нажмите Зарегистрироваться". Below the text are two input fields: the first is labeled "Имя пользователя" and the second is labeled "Пароль". A "Зарегистрироваться" button is positioned below the password field.

Рис. 22. Страница регистрации в системе.

Интерфейс системы регистрации работает схожим образом с системой авторизации, с разницей в том, что выдаются немного другие сообщения на действия пользователей. После регистрации система отправит пользователя на экран авторизации для дальнейшей авторизации в системе.

После авторизации и отправки пользователя на главный экран, который показан на рис. 23, пользователю становится доступен основной функционал сайта, с которым предстоит работать пользователям.

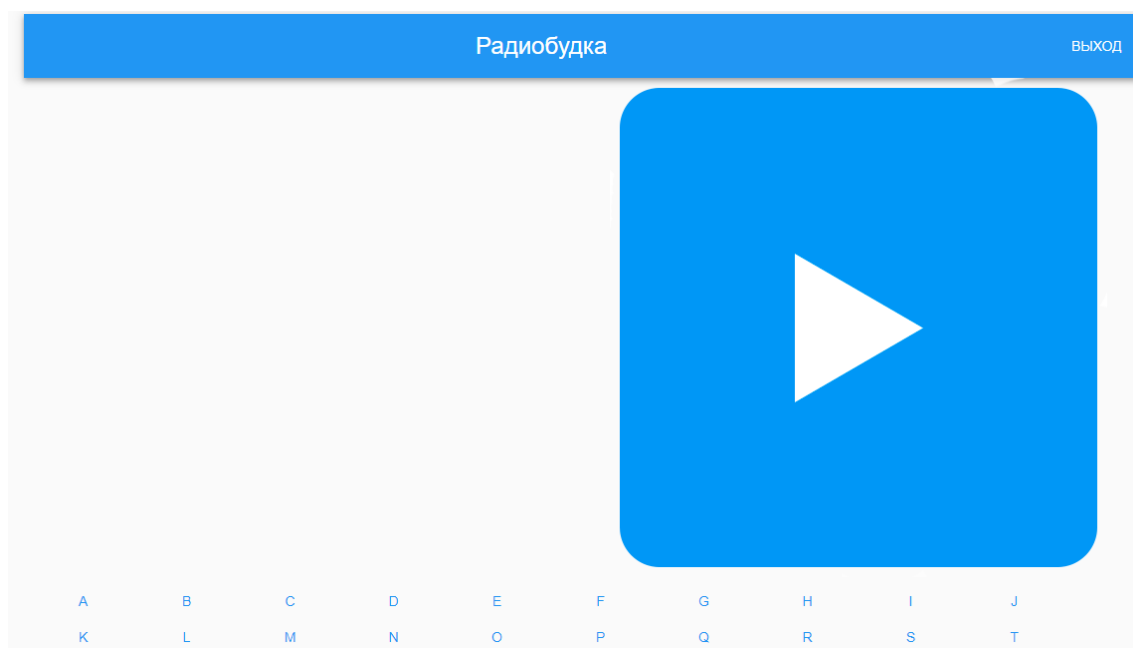


Рисунок 23. Интерфейс авторизованного пользователя

На главном экране системы, как видно на рис. 24, есть несколько важных элементов, во-первых, это кнопка запуска играющего списка композиций, во-вторых, присутствует набор кнопок, с помощью этого набора кнопок можно фильтровать наборы аудиозаписей, которые можно добавить в плей-лист, тем самым предложив ее к прослушиванию. На данном рисунке, не выбрано ни одной аудиозаписи, поэтому в плей-листе нет музыкальных композиций. После нажатия кнопки запуска включается актуальная аудиозапись.

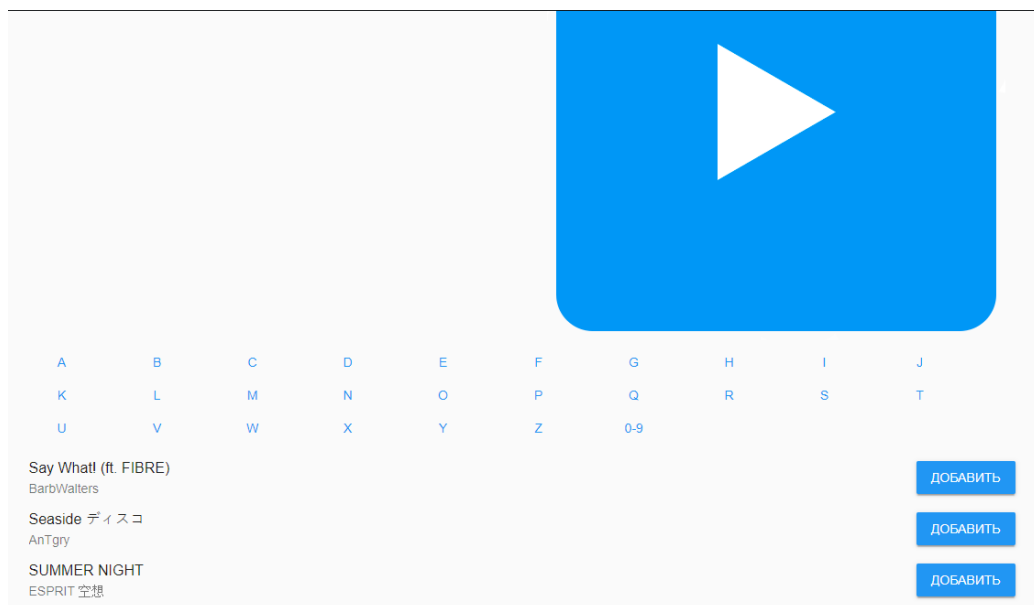


Рис. 24. Элемент интерфейса для добавления аудиозаписей

После нажатия на одну из кнопок с буквой, открывается список все аудиозаписей в базе данных, начинающихся с буквы, которая была указана на нажатой кнопке. У предложенных аудиозаписей указаны название и автор, также присутствует кнопка добавить которая отправляет запись в плей-лист, для дальнейшего воспроизведения. Стоит отметить, что при заказе аудиозаписи пользователь лишается права голосовать или добавлять новые записи, пока предложенная им запись, или запись, за которую он про голосовал находится в плей-листе.

При заполненном плей-листе, как показано на рис. 25, аудиозаписи автоматически переключаются по истечении времени воспроизведения. Время, которое осталось до конца аудиозаписи указано в правом верхнем углу рядом с кнопкой Выход. Сама аудиозапись находится с правой стороны по середине. Плей-лист распределяет очередь аудиозаписей по двум критериям: количеству голосов, отданных записи с аудиозаписью от пользователей, и времени заказа аудиозаписи. Причем количество голосов является приоритетным критерием при распределении аудиозаписей в плей-листе.

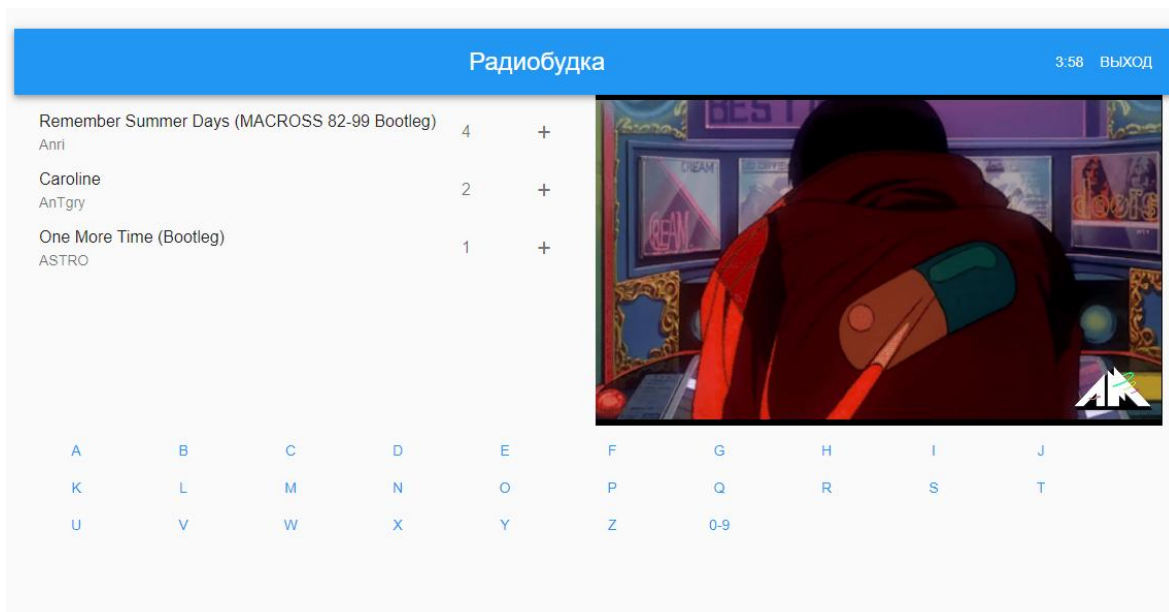


Рис. 25. Полностью задействованный главный экран

Также есть система для администрирования базы данных веб-сайта, сформированная фреймворком Django. Чтобы попасть в панель администрирования нужно зайти по адресу сервера Django с припиской /admin. После чего, как продемонстрировано на рис. 26, открывается панель для входа в систему управления сервером Django.

Администрирование Django

Имя пользователя:

Пароль:

Войти

Рис. 26. Форма входа в панель администрирования Django

После ввода данных суперпользователя, сайт отправляет администратора, как видно из рис. 27, на страницу со всеми таблицами созданными на основе классов, описанных в блоке Models.

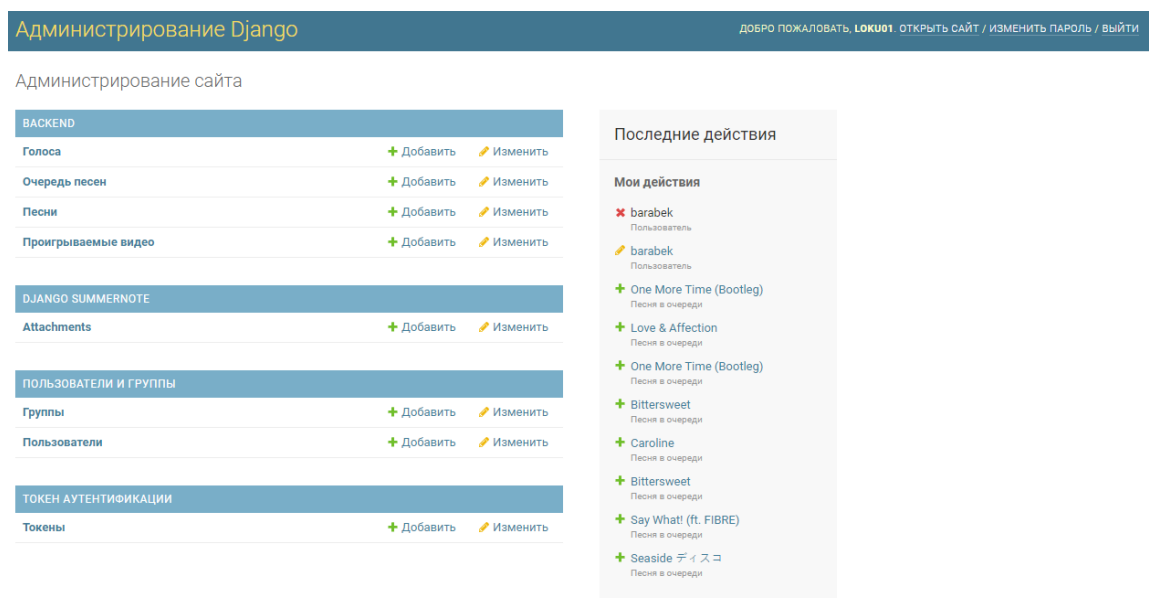


Рис. 27. Страница с таблицами базы данных в интерфейсе администратора Django

В панели администратора представлены все таблицы базы данных, во всех таблицах базы данных можно просматривать, изменять, добавлять и удалять данные. Так же, через эту панель можно изменять права доступа пользователей, что может пригодиться при дальнейшей разработке системы.

Также в рамках реализации информационной системы была разработана структура базы данных, которая показана на рис. 28, на основе которой строились модели в Django.

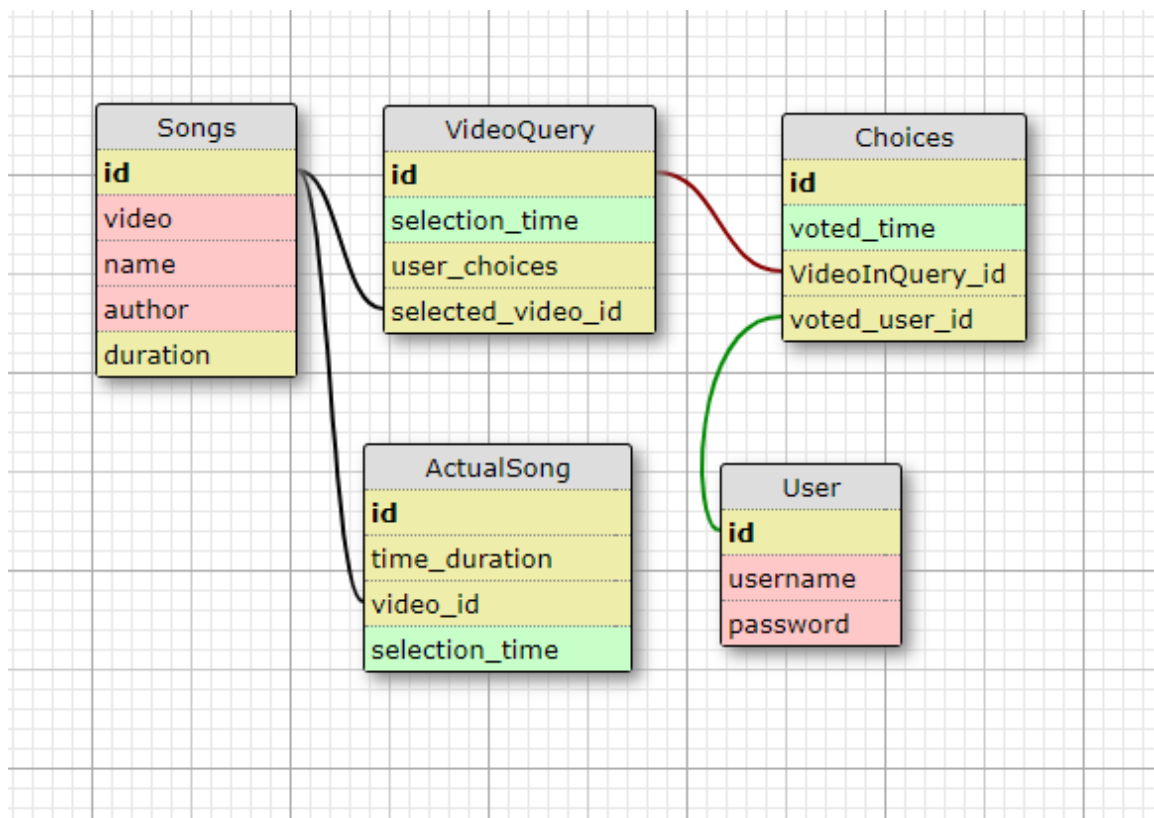


Рис. 28. Структура базы данных, разработанного проекта

Таким образом все поставленные задачи реализованы в полной мере. Результатом данной работы является локальная версия «Автоматизированной информационной системы сетевой радиостанции». Развертывание системы на сервере не производилась.

2.3 Результаты апробации

По окончании разработки и после согласования с научным руководителем Стариченко Б.Е. было решено провести апробация информационной системы в формате экспертной оценки. Экспертами выступали студенты ИМФИиТ.

Экспертам был выдан доступ к сайту и было проведено базовое обучение основам пользования сайтом.

После работы с сайтом экспертам был выдан доступ к странице с опросом, организованным средствами сайта Google Docs. Шаблон опроса представлен в ПРИЛОЖЕНИИ 1.

Как видно из рис. 29, экспертами было отмечено, что разработанная информационная система удовлетворяет требованиям, заявленным при описании технического задания.

Удовлетворяет ли информационная система требования технического задания

6 ответов

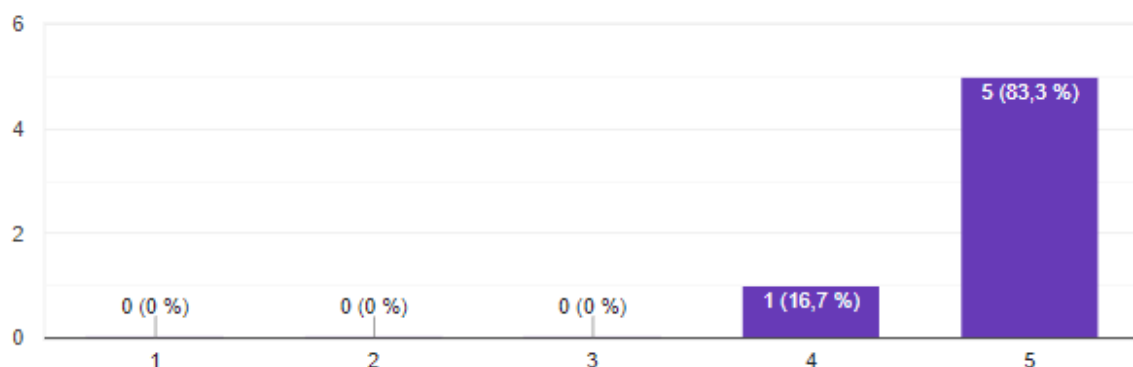


Рис. 29. Результаты опроса по вопросу удовлетворительности исполнения технического задания.

Также в ходе анкетирования как указано на рис. 30, было выявлено что разработанный интерфейс является удобным для эксплуатации, и удовлетворяет большинство опрошенных.

Удобно ли пользоваться интерфейсом

6 ответов

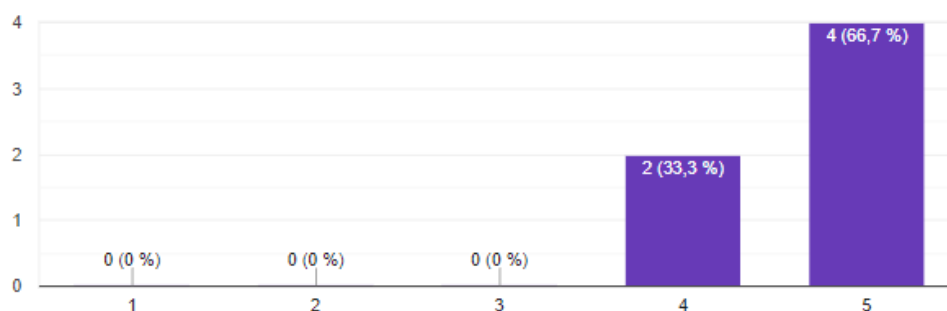


Рис. 30. Результаты опроса по вопросу удобства пользования интерфейсом.

Заключение

Результатом выпускной квалификационной работы является разработка информационной системы «Автоматизированной информационной системы сетевой радиостанции» в соответствии с техническим заданием. Система автоматически организует список музыкальных аудиозаписей посредством выбора пользователей. В ходе разработки были реализованы следующие задачи:

1. Произведен анализ технологий реализации сетевой радиостанции, выявлена специфика автоматизированной информационной системы сетевой радиостанции.
2. Проанализирован и обоснован выбор аппаратно-программной платформы, определить структуру, объекты web-интерфейса автоматизированной информационной системы сетевой радиостанции.
3. В соответствии с техническим заданием смоделирована, спроектирована и реализована автоматизированная информационная система сетевой радиостанции.
4. Проведена апробация.

Разработанная информационная система удовлетворяет техническому заданию. Таким образом, можно утверждать, что цели выпускной квалификационной работы достигнуты, задачи выполнены в полном объёме.

Список информационных источников

1. «Как бы радио». Но как бы нет // «Журналист» URL: <https://jrnlst.ru/podcasts-nikulicheva> (дата обращения: 20.01.2019).
2. «Как это работает»: радио — конец прекрасной эпохи // HABR URL: <https://habr.com/ru/company/audiomania/blog/401393/> (дата обращения: 20.01.2019).
3. База данных // HOSTiQ URL: <https://hostiq.ua/wiki/database/> (дата обращения: 20.01.2019).
4. ЖИЗНЕННЫЙ ЦИКЛ ИНФОРМАЦИОННОЙ СИСТЕМЫ // sites.google.com/site/anisimovkhv
5. Интернет // Википедия URL: <https://ru.wikipedia.org/wiki/Интернет> (дата обращения: 20.01.2019).
6. Искусственная нейронная сеть // Википедия URL: https://ru.wikipedia.org/wiki/Искусственная_нейронная_сеть (дата обращения: 20.01.2019).
7. Как начиналось интернет-радиовещание в мире и в России // HABR URL: <https://habr.com/ru/post/288944/> (дата обращения: 20.01.2019).
8. Незаконное распространение контента в сети: ищем правых и виноватых // РАПСИ Российское агентство правовой и судебной информации URL: http://rapsinews.ru/judicial_analyst/20101227/251430986.html (дата обращения: 20.01.2019).
9. Нестерова Н.Г. ИНТЕРНЕТ-РАДИО КАК НОВЫЙ ТИП МЕДИА И ОСНОВА НОВЫХ ДИСКУРСИВНЫХ ПРАКТИК // Вестник Томского государственного университета. - 2013. - №376. - С. 20-24.
10. Плей-лист // Википедия URL: <https://ru.wikipedia.org/wiki/Плей-лист> (дата обращения: 20.01.2019).
11. Подкастинг // Википедия URL: <https://ru.wikipedia.org/wiki/Подкастинг> (дата обращения: 20.01.2019).

12. ПОЧЕМУ МОБИЛЬНОЕ FM-РАДИО УХОДИТ В ИСТОРИЮ? // ITквариат
URL: <https://itkvariat.com/news/1482-pochemu-mobilnoe-fm-radio-uhodit-v-istoriju.html> (дата обращения: 20.01.2019).
13. Радио // Википедия Свободная энциклопедия URL:
<https://ru.wikipedia.org/wiki/Радио> (дата обращения: 20.01.2019).
14. Радиовещание // Википедия URL: <https://ru.wikipedia.org/wiki/Радиовещание>
(дата обращения: 20.01.2019).
15. Радиоприемник // Википедия URL:
<https://ru.wikipedia.org/wiki/Радиоприёмник> (дата обращения: 20.01.2019).
16. Радиостанция // Википедия Свободная энциклопедия URL:
<https://ru.wikipedia.org/wiki/Радиостанция> (дата обращения: 20.01.2019).
17. Управление перспективных исследовательских проектов Министерства обороны США // Википедия URL:
https://ru.wikipedia.org/wiki/Управление_перспективных_исследовательских_проектов_Министерства_обороны_США (дата обращения: 20.01.2019).
18. Фреймворк // Википедия URL: <https://ru.wikipedia.org/wiki/Фреймворк> (дата обращения: 20.01.2019).
19. Что такое виртуальное окружение? Зачем оно? Какие бывают аналоги? // Devman URL: <https://devman.org/qna/12/chto-takoe-virtualnoe-okruzhenie-zachem-ono-kakie-byvajut-analogi/> (дата обращения: 20.01.2019).
- 20.3 Key Software Principles You Must Understand // Envato Tuts URL:
<https://habr.com/ru/post/144611/> (дата обращения: 20.01.2019).
21. Angular (фреймворк) // Википедия URL:
[https://ru.wikipedia.org/wiki/Angular_\(фреймворк\)](https://ru.wikipedia.org/wiki/Angular_(фреймворк)) (дата обращения: 20.01.2019).
22. API // Википедия URL: <https://ru.wikipedia.org/wiki/API> (дата обращения: 20.01.2019).
23. Apple // Википедия URL: <https://ru.wikipedia.org/wiki/Apple> (дата обращения: 20.01.2019).

24. ARPANET // Википедия URL: <https://ru.wikipedia.org/wiki/ARPANET> (дата обращения: 20.01.2019).
25. Front end и back end // Википедия URL: https://ru.wikipedia.org/wiki/Front_end_и_back_end (дата обращения: 20.01.2019).
26. iPod // Википедия URL: <https://ru.wikipedia.org/wiki/iPod> (дата обращения: 20.01.2019).
27. iTunes // Википедия URL: <https://ru.wikipedia.org/wiki/iTunes> (дата обращения: 20.01.2019).
28. JavaScript // Википедия URL: <https://ru.wikipedia.org/wiki/JavaScript> (дата обращения: 20.01.2019).
29. Meet Django // Django URL: <https://www.djangoproject.com> (дата обращения: 20.01.2019).
30. Node.js // Википедия URL: <https://ru.wikipedia.org/wiki/Node.js> (дата обращения: 20.01.2019).
31. NSFNet // Википедия URL: <https://ru.wikipedia.org/wiki/NSFNet> (дата обращения: 20.01.2019).
32. PyCharm // JetBrains URL: <https://www.jetbrains.com/pycharm/> (дата обращения: 20.01.2019). URL: <https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema2> (дата обращения: 20.01.2019).
33. React // Википедия URL: <https://ru.wikipedia.org/wiki/React> (дата обращения: 20.01.2019).
34. URL // Википедия URL: <https://ru.wikipedia.org/wiki/URL> (дата обращения: 20.01.2019).
35. Vue.js // Википедия URL: <https://ru.wikipedia.org/wiki/Vue.js> (дата обращения: 20.01.2019).
36. Welcome to Python.org // Python URL: <https://www.python.org> (дата обращения: 20.01.2019).

37. Why use Node.js // ThinkMobiles URL: <https://thinkmobiles.com/blog/why-use-nodejs/> (дата обращения: 20.01.2019).
38. Why We Use Django Framework & What Is Django Used For // Djangostars URL: <https://djangostars.com/blog/why-we-use-django-framework/> (дата обращения: 20.01.2019).
39. YouTube // Википедия URL: <https://ru.wikipedia.org/wiki/YouTube> (дата обращения: 20.01.2019).
40. YouTube Music // Википедия URL: https://ru.wikipedia.org/wiki/YouTube_Music (дата обращения: 20.01.2019).

Приложения

Приложение 1.

Шаблон анкеты опроса по работе с сайтом

1. Удовлетворяет ли информационная система требования
технического задания:
Не удовлетворяет (1-5) Удовлетворяет
2. Соответствует ли приложение формату автоматизированной
информационной системы сетевой радиостанции:
Не соответствует (1-5) Соответствует
3. На сколько понятным выглядит интерфейс:
Не понятен (1-5) Понятен
4. Удобно ли пользоваться интерфейсом:
Не удобно (1-5) Удобно